

How to run ASAP & MST Slicing

N. Saoulidou, Fermilab, 11-18-04

Outline

- ASAP & MST Slicing in the offline software
 - How to select them to run.
 - Parameter description for each method.
- Summary & On-going work

Quick Description of methods :ASAP

- ASAP slicing is based on the most simple (**A**s **S**imple **A**s **P**ossible) idea one could come up with:
 - Sort strips by begin time and then form a difference slice whenever you find a time difference greater than a cut!
- Parameters of this method (that are all tunable) :
 - Minimum PH of strip in order to be considered as a point
 - Minimum number of strips to form a slice
 - Time difference above which you start a new slice.
- This method forms slices anywhere (calorimeter and spectrometer). The spectrometer slices are never of course reconstructed due to the 4-fold PLEX.

Quick Description of methods :MST

- MST slicing is based on a clustering technique, from Graph Theory, called **Minimum Spanning Trees**. It groups together points (strips) using a metric that is :
 - **Time difference + A * Space difference**
- Currently the space difference is difference in z.
- Parameters of this method (that are all tunable) :
 - **Minimum PH of strip in order to be considered as a point**
 - **Minimum number of strips to form a slice**
 - **A, (Participation of the space difference in the metric)**
 - **Length, above which you cut the branch to form slices.**
- This method forms slices anywhere (calorimeter and spectrometer). The spectrometer slices are never of course reconstructed due to the 4-fold PLEX.

How to Run them & Tune the parameters 1.

- To select MST slicer do (in the job script):

```
AlgHandle ah = af.GetAlgHandle("AlgSliceSRList","FarBeam");  
(or NearBeam )
```

```
AlgConfig &ac = ah.GetAlgConfig();
```

```
ac.UnLockValues();
```

```
ac.Set("passMST",1)
```

```
ac.LockValues();
```

- Tunable Parameters for MST:

- **Double_t MinCharge** (minimum strip PH in PE's)
- **Int_t MinStrip** (minimum # of strip to form a slice)
- **Double_t MaxLen** (maximum branch length to start a new slice)
- **Double_t Zfact** (Weight of the z difference in the metric definition)

How to Run them & Tune the parameters 2.

- To select ASAP slicer do (in the job script):

```
AlgHandle ah = af.GetAlgHandle("AlgSliceSRList","FarBeam");  
(or NearBeam )
```

```
AlgConfig &ac = ah.GetAlgConfig();
```

```
ac.UnLockValues();
```

```
ac.Set("passASAP,1)
```

```
ac.LockValues();
```

- Tunable Parameters for ASAP:

- **Double_t MinCharge** (minimum strip PH in PE's)
- **Int_t MinStrip** (minimum # of strip to form a slice)
- **Double_t NBuck** (maximum time difference in buckets = 19 ns to start a new slice)

Performance & Comparison with SR

- So far I have performed comparisons between MST & ASAP with SR and the results I have present are summarized as follows:
 - MST Performs as good as the SR (Maybe slightly better) for both High and Low intensity Near Events
 - ASAP Performs slightly worse than the SR for low intensity Near Events.
- But now that R1.12 is tagged and the production of the MC files will start, I will have :
 - 1. **More statistics**
 - 2. **More stable conditions as far as code changes are concerned**
 - 3. **Less code failures-crashes-e.t.c that occur when I run with my own test-release using development that take up too much time.**

which means I will be able to perform a final. more reliable comparison between the different techniques.

Potential of Both methods

- Both methods are relatively simple (or very simple) with a low number of parameters.
- The MST slicing has a potential to be developed further, improved & tuned better.
- The ASAP I don't think has a great potential of developing further. However is a simple tool that can become loose or strict very easily as far as slicing is concerned, and serve different physics analysis requirements or used a "second" step to make slices more clear.
- The SR is performing just fine! (all methods have advantages and disadvantages) I just have a personal preference on simpler code (that performs as well) with low number of tunable parameters that I can easily modify.

Summary & Ongoing work

- I added both slicing methods in the offline software so that users can easily select (or further develop/change) any of the existing 3 (SR - MST - ASAP) .
- All parameters of my code are tunable.
- I can now more easily use the methods I wrote, to perform a clean final comparison of the different slicing techniques under stable conditions and with higher statistics.
- Also I can more easily & quickly perform a systematic tuning of the parameters (using the MDC candidate files and frozen R1.12 instead of development, that is by definition unstable while developing).
- I plan to have more on comparison/tuning next time.